

JAVA / J2EE / JUMP

Q: How does Microsoft compete against J2EE?

A: J2EE, and admittedly, Windows DNA, were 2nd generation platforms. Microsoft .NET represents the 3rd generation, which means it a) uses public Internet standards to integrate with many platforms; b) was designed specifically for the Internet to literally make the Internet a programmable platform; and c) embraces the way developers work today, for example, by supporting over 20 different programming languages. J2EE offers none of these benefits.

Q: Will anyone be providing a Java implementation for .NET?

A: Microsoft announced an implementation in January 2001, called JUMP. Java User Migration Path to Microsoft .NET (JUMP to .NET) combines technology and service offerings that enable programmers to preserve, enhance, and migrate Java language projects onto the Microsoft .NET platform. JUMP to .NET enables Microsoft Visual J++® customers and other programmers who use the Java language to take advantage of existing skills and code investments while fully exploiting the Microsoft platform today and into the future. JUMP to .NET provides the easiest transition for Java developers into the world of XML-based Web Services, and dramatically improves the interoperability of the Java language with software written in a variety of other programming languages. More information can be found on <http://msdn.microsoft.com/visualj>.

Q: But your JUMP support is JDK 1.1.4 only - the real action is on J2, isn't it?

A: Hardly. Our research shows that most Java development is 1.1.4 or earlier, and certainly all of the Visual J++ code is 1.1.4. Our responsibility with JUMP is first and foremost to VJ customers, and we think there's a lot of value for anyone moving from Java to .NET. At the end of the day, it's a platform choice, and we're very confident that .NET will win.

Q: Isn't C# just a knockoff of Java?

A: No. C# is a simple modern component oriented language that is an evolution of C and C++. In developing C#, we address the needs of a certain class of developers who want simplicity and power in a managed environment. Additionally C# innovates in several areas of language design. For example, C# deals with problems like component Versioning and Platform Integration in elegant and innovative ways.

Q: How does .NET Framework security compare with Java?

A: First, .NET Framework security was built-in from the start, not added later. Code access security provides a more powerful and granular infrastructure for security enforcement, with permissions that can be extended by customer code to protect their resources. Evidence-based security policy allows administrators to precisely control what code is assigned what permission. User identity classes unify various forms of authentication into a single model, and customers can add to the model with their own.

Q: What's the difference between the Microsoft Common Language Runtime for .NET and the Java virtual-machine?

A: The .NET Framework is different from Java, down to its fundamental design goals. The .NET Framework was designed to be a high-performance, high-productivity development and execution environment. Java was designed to provide a least-common-denominator set of services across different operating systems. These design differences manifest themselves throughout the architecture. For example, the .NET Framework is a multi-programming language environment: we have more than 20 language partners who are targeting the .NET Framework. By giving developers a choice of programming languages, we are enabling them to choose the right language for the job at hand or to leverage knowledge of an existing language. Java was designed for a single programming language. The .NET Framework was designed for performance, so it executes native code. Java was designed to be interpreted, and only was it later retrofitted with the capability to compile down to native code.

Q: What are the main differentiators between the SUN/Java initiative and Microsoft.NET?

A: It is never clear what “Java” means anymore. If you were referring to Java the platform, then the .NET Framework technology would be competitive at some high level of abstraction, just like any other environment that includes a set of APIs such as Windows DNA. But the .NET technology is fundamentally different and it is inaccurate to pigeonhole it as a competitive alternative to Java. The most obvious difference is that it can consume multiple programming languages. There are many other equally significant, fundamental differences between the technologies. For example, XML is integral to the platform and is seen in several key technologies including XML Web services.